JSAI : Journal Scientific and Applied Informatics
Vol. 08, No. 2, Juni 2025, hal. 427~436
E-ISSN: 2614-3054; P-ISSN: 2614-3062, accredited by Kemenristekdikti, Sinta 4
DOI: 10.36085

Perbandingan Performa Algoritma *Random Tree*, K-NN, dan A-NN untuk Deteksi Serangan DDoS pada *Software Defined Network* (SDN)

[1]Akbar Pandu Segara, [2]Muhammad Andryan Wahyu Saputra, [3]Narandha Arya Ranggianto
[1,2,3]Universitas Jember, Indonesia
[1]akbarpandu@unej.ac.id; [2]andryan@unej.ac.id; [3]ranggi@unej.ac.id

Article Info

Article history:

Received, 2025-06-04
Revised, 2025-06-13
Accepted, 2025-06-19

Kata Kunci:

software denifed network
ddos
pembelajarn mesin
Random Tree
K-Nearest Neighbor
Artificial Neural Network

ABSTRAK

Software Defined Network (SDN) dengan arsitektur terpusat memiliki kerentanan terhadap serangan *Distributed Denial of Service* (DDoS) yang dapat menyebabkan kegagalan layanan jaringan secara menyeluruh. Penelitian ini bertujuan untuk membandingkan performa tiga algoritma *Machine Learning* yaitu *K-Nearest Neighbor* (K-NN), *Artificial Neural Network* (ANN), dan *Random Tree* dalam mendeteksi serangan DDoS pada lingkungan SDN. Dataset DDoS-SDN yang terdiri dari 104.345 baris dan 23 kolom digunakan dengan pembagian data 70% untuk pelatihan dan 30% untuk pengujian. Evaluasi dilakukan menggunakan metrik akurasi, presisi, recall, F1-score, dan AUC-ROC. Hasil penelitian menunjukkan bahwa A-NN memperoleh performa terbaik dengan akurasi 96,85%, presisi 94,35%, recall 97,79%, F1-score 96,04%, dan AUC 0,994, diikuti oleh K-NN dengan akurasi 88,89% dan *Random Tree* dengan akurasi terendah 86,49%. Keunggulan A-NN disebabkan oleh kemampuannya menangkap pola non-linear kompleks, melakukan ekstraksi fitur otomatis, dan beradaptasi terhadap heterogenitas data dari 22 fitur yang digunakan. Temuan ini menunjukkan bahwa A-NN merupakan pilihan optimal untuk implementasi sistem deteksi serangan DDoS real-time dalam lingkungan SDN, memberikan dasar yang kuat untuk pengembangan sistem keamanan jaringan berbasis pembelajaran mesin yang cerdas dan adaptif.

Keywords:

software denifed network
ddos
Machine Learning
Random Tree
K-Nearest Neighbor
Artificial Neural Network

ABSTRACT

Software-Defined Networks (SDNs) with a centralized architecture are vulnerable to Distributed Denial of Service (DDoS) attacks, which can cause widespread network service failures. This study aims to compare the performance of three Machine Learning algorithms—K-Nearest Neighbor (K-NN), Artificial Neural Network (ANN), and Random Tree—in detecting DDoS attacks in an SDN environment. The DDoS-SDN dataset, consisting of 104,345 rows and 23 columns, was used with a data split of 70% for training and 30% for testing. Evaluation was conducted using accuracy, precision, recall, F1-score, and AUC-ROC metrics. The results showed that ANN achieved the best performance with an accuracy of 96.85%, precision of 94.35%, recall of 97.79%, F1-score of 96.04%, and AUC of 0.994, followed by K-NN with an accuracy of 88.89% and Random Tree with the lowest accuracy of 86.49%. The superiority of ANN is attributed to its ability to capture complex non-linear patterns, perform automatic feature extraction, and adapt to the heterogeneity of data from the 22 features used. These findings indicate that ANN is the optimal choice for implementing a real-time DDoS attack detection system in an SDN environment, providing a strong foundation for the development of intelligent and adaptive Machine Learning-based network security systems.

Penulis Korespondensi:
Akbar Pandu Segara,
Program Studi Teknologi Informasi,
Universitas Jember,
Email: akbarpandu@unej.ac.id

1. PENDAHULUAN

Dalam era digital yang semakin berkembang pesat, teknologi jaringan komputer memainkan peran krusial dalam mendukung komunikasi data yang handal, efisien, dan adaptif. Salah satu inovasi signifikan dalam bidang jaringan adalah pengenalan paradigma *Software Defined Network* (SDN). SDN merupakan pendekatan arsitektur yang memisahkan antara *control plane* (logika pengendali jaringan) dan *data plane* (komponen pengirim data), yang sebelumnya terintegrasi dalam perangkat jaringan konvensional. Pemisahan ini memungkinkan kontrol jaringan dilakukan secara terpusat melalui *controller*, serta menjadikan jaringan lebih fleksibel dan mudah dikelola [1]. Dengan meningkatnya adopsi SDN dalam berbagai sektor seperti cloud computing, *Internet of Things* (IoT), tantangan keamanan juga ikut berkembang. Arsitektur terpusat SDN memang membawa kemudahan pengelolaan, tetapi pada saat yang sama menciptakan titik kegagalan tunggal (*single point of failure*) yang sangat rentan terhadap serangan siber, khususnya serangan *Distributed Denial of Service* (DDoS). Dalam skenario ini, controller SDN dapat dibanjiri oleh lalu lintas palsu atau permintaan yang tidak sah dalam jumlah besar, yang menyebabkan ketidakmampuan *controller* dalam menangani lalu lintas jaringan yang sah, sehingga menyebabkan kegagalan layanan jaringan secara menyeluruh[2].

Serangan DDoS pada lingkungan SDN memiliki karakteristik yang berbeda dibandingkan dengan jaringan tradisional karena sifat terpusat dari kontrol jaringan. Hal ini memerlukan pendekatan deteksi dan mitigasi yang lebih adaptif, *real-time*, dan cerdas. Dalam konteks ini, pendekatan berbasis *Machine Learning* (ML) telah menunjukkan potensi besar sebagai solusi untuk mendeteksi dan mengklasifikasi lalu lintas jaringan secara otomatis guna mengidentifikasi serangan DDoS sejak dini. Teknik ML mampu menganalisis pola lalu lintas jaringan secara cepat, mengenali anomali, serta menyesuaikan model berdasarkan pembelajaran dari data historis [3].

Ketiga buah algoritma *Machine Learning* yang cukup populer dan banyak diterapkan dalam konteks deteksi serangan jaringan adalah *K-Nearest Neighbor* (K-NN), *Artificial Neural Network* (A-NN), dan *Random Tree* (RT). Masing-masing algoritma memiliki keunggulan dan keterbatasan. Algoritma K-NN dikenal dengan pendekatan berbasis instance, di mana klasifikasi dilakukan berdasarkan kedekatan fitur antar data. K-NN tidak memerlukan fase pelatihan kompleks dan sangat efektif untuk dataset kecil hingga menengah. Dalam studi oleh [4], [5], [6]. K-NN menunjukkan performa yang baik dalam mengidentifikasi anomali lalu lintas jaringan dengan tingkat akurasi yang cukup tinggi.

Sementara itu, A-NN, sebagai representasi jaringan saraf buatan, memiliki kemampuan untuk mengenali pola yang kompleks dan non-linear. A-NN dapat digunakan untuk klasifikasi maupun regresi, dan telah terbukti efektif dalam mendeteksi berbagai serangan jaringan, meskipun membutuhkan sumber daya komputasi yang lebih besar serta waktu pelatihan yang lebih lama. Pada penelitian [7] menunjukkan bahwa A-NN berhasil meningkatkan deteksi serangan DDoS dalam lingkungan jaringan virtual dengan akurasi lebih dari 95%, namun dengan kebutuhan tuning parameter yang cukup kompleks. Di sisi lain, *Random Tree* (dan variannya Random Forest) adalah algoritma berbasis *ensemble learning* yang menggabungkan beberapa pohon keputusan untuk meningkatkan akurasi dan stabilitas model. Algoritma ini mampu menangani data dengan dimensi tinggi dan menangkap interaksi kompleks antar fitur. Dalam penelitian [8] menemukan bahwa Random Forest mampu mencapai akurasi hingga 97% dalam mendeteksi serangan DDoS pada SDN menggunakan dataset DDoS SDN dataset, dan outperforming algoritma lain dalam hal kecepatan dan akurasi.

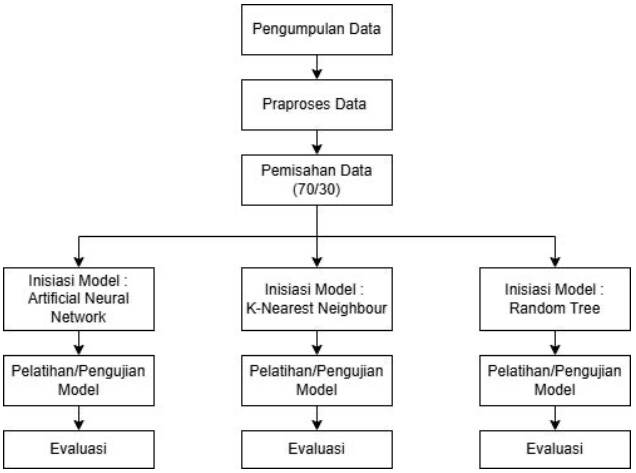
Namun demikian, masih terdapat kesenjangan dalam literatur ilmiah terkait perbandingan menyeluruh performa antara K-NN, A-NN, dan *Random Tree* dalam konteks deteksi DDoS pada arsitektur SDN. Banyak studi hanya menyoroti satu atau dua algoritma tanpa memberikan perspektif komparatif yang utuh. Padahal, dalam implementasi nyata, pemilihan algoritma yang tepat sangat bergantung pada kebutuhan spesifik seperti akurasi, waktu pelatihan, waktu inferensi, serta kompleksitas model. Oleh karena itu, analisis komparatif menjadi penting untuk memberikan wawasan yang lebih mendalam mengenai kekuatan dan kelemahan masing-masing pendekatan dalam lingkungan jaringan yang dinamis seperti SDN. Selain itu, penting juga untuk menggunakan dataset yang representatif dan simulasi kondisi nyata dalam pengujian performa algoritma agar hasil yang diperoleh dapat digeneralisasi. Beberapa dataset publik seperti CICIDS2017, NSL-KDD, dan dataset khusus SDN telah banyak digunakan dalam penelitian terdahulu. Studi oleh [3], [9], [10] juga menyarankan penggunaan teknik *preprocessing* yang tepat seperti normalisasi data dan reduksi dimensi untuk meningkatkan akurasi model *Machine Learning*.

Dengan mempertimbangkan urgensi masalah keamanan pada SDN serta potensi besar algoritma *Machine Learning* dalam mendeteksi ancaman, maka penelitian ini bertujuan untuk melakukan analisis perbandingan performa antara algoritma K-NN, A-NN, dan *Random Tree* dalam mendeteksi serangan DDoS pada lingkungan *Software Defined Network*. Penelitian ini akan mengukur performa algoritma berdasarkan metrik akurasi, precision, recall, dan F1-score, serta mengevaluasi efisiensi masing-masing model dalam hal waktu pelatihan dan prediksi. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi ilmiah dalam pemilihan metode

deteksi yang optimal untuk penerapan keamanan pada infrastruktur SDN, sekaligus menjadi referensi bagi pengembang sistem deteksi dini serangan berbasis pembelajaran mesin.

2. METODE PENELITIAN

Pada penelitian ini terdapat kerangka yang menggambarkan tahapan-tahapan yang dilakukan dalam proses pengembangan dan evaluasi model deteksi serangan DDoS. Dimulai dari tahap pengumpulan data, kemudian data dilakukan praproses untuk memastikan kualitas data yang baik. Setelah itu, data dibagi menjadi dua bagian, yaitu 70% untuk pelatihan dan 30% untuk pengujian. Tahapan selanjutnya adalah inisialisasi dan pelatihan tiga jenis model klasifikasi yang berbeda, yaitu *Artificial Neural Network*, *K-Nearest Neighbour*, dan *Random Tree*. Masing-masing model kemudian dievaluasi menggunakan metrik performa klasifikasi untuk menentukan model terbaik. Pada gambar 1 akan menyajikan keseluruhan proses tersebut secara sistematis.



Gambar 1. Kerangka Penelitian Deteksi Serangan DDoS

A. Pengumpulan Data

Penelitian ini menggunakan dataset *DDoS-SDN* yang tersedia secara publik melalui platform Kaggle. Dataset ini dipilih karena relevansinya yang tinggi dengan topik penelitian, yaitu deteksi serangan DDoS pada lingkungan *Software Defined Network* (SDN), serta kualitas data yang telah tervalidasi untuk keperluan penelitian akademik. Dataset ini memiliki ukuran 104.345 baris dan 23 kolom dalam format CSV (Comma Separated Values) yang dapat diakses dengan lisensi publik untuk keperluan penelitian akademik. Fitur target dalam dataset adalah Label yang menggunakan sistem pelabelan biner, dimana nilai 0 merepresentasikan traffic jaringan normal (kelas 'Normal') dan nilai 1 merepresentasikan serangan DDoS (kelas 'DDoS').

Sistem pelabelan biner ini dirancang khusus untuk memudahkan implementasi pendekatan supervised learning dalam pelatihan model deteksi serangan. Pelabelan yang akurat memungkinkan algoritma *Machine Learning* untuk mempelajari pola yang membedakan antara traffic normal dan serangan DDoS secara efektif. Hal ini menjadikan dataset sangat cocok untuk evaluasi dan perbandingan performa berbagai algoritma seperti *K-Nearest Neighbor*, *Artificial Neural Network*, dan *Random Tree* dalam konteks deteksi serangan DDoS.

Keunggulan utama dataset ini terletak pada fitur-fitur yang spesifik untuk lingkungan SDN, seperti jumlah pesan OpenFlow packet-in, informasi tentang flow management, dan data interaksi switch-controller. Granularitas data di level flow memberikan detail monitoring yang tinggi, sementara fitur timestamp memungkinkan analisis karakteristik real-time. Dataset ini berhasil mencapture berbagai aspek perilaku jaringan yang penting untuk deteksi anomali, menjadikannya sangat sesuai untuk penelitian deteksi serangan DDoS pada *Software Defined Network* dengan tingkat akurasi dan detail yang tinggi.

Dataset yang digunakan terdiri dari 23 fitur yang merepresentasikan berbagai parameter penting dalam lalu lintas jaringan pada lingkungan SDN. Fitur-fitur temporal dan identifikasi meliputi timestamp (Dt) dan Switch ID yang menunjukkan waktu dan lokasi pengumpulan data. Fitur alamat dan komunikasi mencakup alamat IP sumber (Src), alamat IP tujuan (Dst), jenis protokol yang digunakan (TCP, UDP, ICMP), dan nomor port jaringan. Aspek volume traffic direpresentasikan melalui jumlah paket (Pktcount), total byte yang ditransfer (Bytecount), serta byte yang dikirim dan diterima (Tx_bytes, Rx_bytes).

Karakteristik durasi dan timing dalam dataset diukur melalui beberapa fitur seperti durasi aliran dalam detik (Dur), durasi dalam nanodetik untuk presisi tinggi (Dur_nsec), dan durasi total untuk agregasi (Tot_dur). Fitur flow dan rate memberikan informasi tentang jumlah aliran dalam periode tertentu (Flows), rata-rata paket per aliran (Pktperflow), rata-rata byte per aliran (Byteperflow), laju pengiriman paket per detik (Pktrate), dan

jumlah aliran bolak-balik antar IP (Pairflow). Khusus untuk lingkungan SDN, dataset menyertakan fitur Packetins yang merepresentasikan jumlah pesan packet-in dari switch ke controller dalam protokol OpenFlow. Fitur bandwidth dalam dataset mencakup laju pengiriman data (Tx_kbps), laju penerimaan data (Rx_kbps), dan total bandwidth gabungan (Tot_kbps) yang diukur dalam kilobit per detik. Untuk penjelasan tentang definisi fitur dapat dilihat pada tabel 1 berikut.

Tabel 1. Nama Fitur

No	Nama Fitur	Definisi
1	Dt	Waktu atau timestamp saat data dikumpulkan
2	Switch	ID atau alamat switch jaringan tempat data dikumpulkan
3	Src (Source Ip address)	Alamat IP pengirim paket
4	Dst (destination IP address)	Alamat IP tujuan paket.
5	Pktcount	Jumlah paket yang dikirim dalam aliran tersebut.
6	Bytecount	Total byte yang ditransfer dalam aliran.
7	Dur (duration)	Lama durasi aliran berlangsung (dalam detik).
8	Dur_nsec (duration per second)	Durasi dalam nanodetik; bisa digunakan untuk menghitung waktu presisi.
9	Tot_dur	Durasi total dari semua aliran, biasanya untuk agregasi.
10	Flows	Jumlah aliran (flows) dalam jangka waktu tertentu.
11	Packetins	Jumlah pesan "packet-in" dari switch ke controller (OpenFlow).
12	Pktperflow	Rata-rata jumlah paket per aliran.
13	Byteperflow	Rata-rata jumlah byte per aliran.
14	Pktrate	Laju pengiriman paket per detik.
15	Pairflow	Jumlah aliran bolak-balik antara sepasang alamat IP.
16	Protocol	Jenis protokol yang digunakan (TCP, UDP, ICMP, dll).
17	Port_no	Nomor port jaringan yang digunakan dalam komunikasi.
18	Tx_bytes	Total byte yang dikirim dari sumber (transmitted bytes).
19	Rx_bytes	Total byte yang diterima oleh tujuan (received bytes).
20	Tx_kbps	Laju penerimaan data dalam kilobit per detik.
21	Rx_kbps	Laju pengiriman data dalam kilobit per detik.
22	Tot_kbps	Total bandwidth (Rx + Tx) dalam kilobit per detik.
23	Label	Jenis kelas dari data

Berdasarkan tabel diatas dataset yang digunakan dalam penelitian ini merepresentasikan berbagai parameter penting dalam lalu lintas jaringan pada lingkungan *Software Defined Network* (SDN). Data dikumpulkan dari lingkungan eksperimen menggunakan pengontrol SDN berbasis OpenFlow (seperti Ryu) yang diintegrasikan dengan *network emulator* Mininet, sehingga mencerminkan kondisi jaringan yang realistis dan dinamis. Dengan pemantauan mendalam terhadap setiap *flow* yang melintas, dataset ini menyediakan informasi yang kaya dan representatif untuk mengidentifikasi lalu lintas normal dan anomali, termasuk serangan DDoS. Karakteristik ini menjadikan dataset sangat sesuai untuk digunakan dalam pelatihan dan pengujian algoritma *Machine Learning* untuk mendeteksi serangan secara akurat dan efisien di lingkungan SDN.

B. Pra Proses Data

Tahap pra proses data merupakan langkah kritis dalam mempersiapkan dataset DDoS-SDN untuk digunakan dalam algoritma *Machine Learning*. Proses ini dimulai dengan eksplorasi data awal untuk memahami karakteristik dan distribusi setiap fitur dalam dataset. Analisis statistik deskriptif dilakukan untuk mengidentifikasi nilai minimum, maksimum, rata-rata, dan standar deviasi dari masing-masing fitur numerik, sementara fitur kategorikal dianalisis untuk memahami distribusi frekuensi dan variasi nilainya.

Penanganan missing values menjadi prioritas utama dalam tahap ini, meskipun dataset DDoS-SDN umumnya memiliki kualitas data yang baik. Setiap kolom diperiksa secara menyeluruh untuk mengidentifikasi adanya nilai yang hilang atau tidak valid. Jika ditemukan missing values, strategi penanganan yang tepat diterapkan berdasarkan karakteristik fitur, seperti imputasi dengan nilai median untuk fitur numerik atau mode untuk fitur kategorikal. Untuk kasus ekstrim dimana missing values terlalu banyak, baris data tersebut akan dihapus untuk menjaga kualitas dataset. Deteksi dan penanganan outliers dilakukan menggunakan metode statistik seperti *Interquartile Range* (IQR) dan Z-score. *Outliers* yang teridentifikasi dievaluasi apakah merupakan anomali yang sah (seperti karakteristik serangan DDoS) atau noise yang dapat mengganggu performa model. *Outliers* yang merupakan bagian dari pola serangan DDoS dipertahankan karena memiliki nilai informatif tinggi, sementara *outliers* yang merupakan kesalahan pengukuran atau noise akan ditangani dengan metode yang sesuai.

Normalisasi dan standarisasi fitur numerik dilakukan untuk memastikan semua fitur memiliki skala yang setara. Mengingat dataset mengandung fitur dengan rentang nilai yang sangat berbeda, seperti *timestamp, packet count*, dan *bandwidth rate*, standarisasi proses menggunakan *Z-score transformation* diterapkan untuk mengkonversi semua fitur numerik ke distribusi dengan mean 0 dan standar deviasi 1. Hal ini penting terutama untuk algoritma *K-Nearest Neighbor* yang sensitif terhadap skala data. *Encoding* fitur kategorikal seperti

protocol type dan *port number* dilakukan menggunakan teknik yang sesuai. *One-hot encoding* diterapkan untuk fitur protokol yang memiliki sedikit kategori unik, sementara label *encoding* dapat digunakan untuk fitur dengan kardinalitas tinggi. Proses ini memastikan bahwa algoritma *Machine Learning* dapat memproses semua jenis fitur dalam dataset secara optimal.

C. Pemisahan Data

Pemisahan data dilakukan menggunakan strategi *train-test split* dengan proporsi 70% untuk data pelatihan dan 30% untuk data pengujian, sesuai dengan praktik standar dalam *Machine Learning* dan konsisten dengan *flowchart* penelitian. *Stratified sampling* diterapkan untuk memastikan distribusi kelas (Normal dan DDoS) pada data pelatihan dan pengujian tetap proporsional dengan distribusi pada dataset asli. Hal ini penting untuk menjaga representativitas kedua kelas dalam proses evaluasi model. Proses pemisahan menggunakan random sampling dengan seed yang tetap untuk memastikan *reproducibility* hasil eksperimen. Penggunaan seed yang konsisten memungkinkan penelitian dapat direplikasi dengan hasil yang identik, sehingga perbandingan antar algoritma dapat dilakukan secara fair dan objektif. Data pelatihan yang terdiri dari sekitar 73.000 sampel akan digunakan untuk melatih ketiga algoritma (K-NN, A-NN, dan *Random Tree*), sementara data pengujian sekitar 31.000 sampel akan digunakan untuk evaluasi performa.

Validasi tambahan dilakukan dengan memeriksa distribusi fitur pada kedua subset untuk memastikan tidak ada bias yang tidak diinginkan. Analisis statistik deskriptif diterapkan pada data pelatihan dan pengujian secara terpisah untuk memverifikasi bahwa kedua subset memiliki karakteristik yang serupa. *Cross-validation strategy* juga dipertimbangkan untuk memberikan estimasi performa yang lebih *robust*, terutama untuk *fine-tuning hyperparameter* pada tahap pelatihan model.

D. Inisialisasi Model

Dalam penelitian ini, digunakan tiga algoritma Machine Learning yang memiliki karakteristik dan pendekatan klasifikasi yang berbeda, yaitu *K-Nearest Neighbor* (K-NN), *Artificial Neural Network* (A-NN), dan *Random Tree*. Pemilihan algoritma ini didasarkan pada kemampuannya dalam melakukan klasifikasi berbasis fitur lalu lintas jaringan, terutama dalam membedakan antara trafik normal dan serangan DDoS pada lingkungan Software Defined Network (SDN). *K-Nearest Neighbor* (K-NN) adalah salah satu algoritma klasifikasi berbasis instance yang mengklasifikasikan suatu data baru berdasarkan jarak terdekat ke sejumlah data pelatihan. Algoritma ini bersifat non-parametrik dan tidak memerlukan pelatihan eksplisit, sehingga sangat cocok untuk data dengan distribusi yang tidak diketahui secara pasti. Dalam konteks deteksi serangan jaringan, K-NN telah terbukti mampu memberikan hasil yang kompetitif dalam mengenali pola anomali berbasis fitur statistik lalu lintas [11], [12]. Namun, kelemahan utama K-NN terletak pada efisiensi waktu dan komputasi saat menangani data berukuran besar, terutama ketika dimensi fitur meningkat.

Salah satu solusi untuk mengatasi tantangan efisiensi dalam pencarian tetangga terdekat pada skala data besar adalah penggunaan *Artificial Neural Network* (A-NN). Terinspirasi dari cara kerja otak manusia, A-NN mampu melakukan pemetaan non-linear dan menangkap pola kompleks dalam data, sehingga dapat membangun representasi data yang lebih terstruktur dan bermakna. Melalui proses pelatihan, A-NN mempelajari struktur laten data dan mengelompokkan titik-titik yang secara semantik berdekatan dalam ruang fitur, memungkinkan pencarian tetangga terdekat dilakukan dengan kompleksitas lebih rendah dibandingkan metode tradisional. Pendekatan ini tidak hanya meningkatkan efisiensi waktu, tetapi juga sangat cocok diterapkan pada sistem berskala besar seperti sistem rekomendasi, deteksi anomali, dan pengenalan pola, serta dapat diintegrasikan dengan berbagai sistem kecerdasan buatan lainnya yang membutuhkan kinerja tinggi dan skalabilitas [13].

Sementara itu, *Random Tree* merupakan algoritma berbasis pohon keputusan yang dibentuk secara acak dari subset fitur dan data latih. Tidak seperti Decision Tree konvensional, *Random Tree* membangun model dengan membagi data secara acak pada tiap simpul untuk membentuk berbagai kemungkinan jalur keputusan. Algoritma ini dikenal karena kemampuannya menangani data dengan fitur kompleks dan bersifat non-linear, serta mampu menghindari *overfitting* yang sering terjadi pada model klasifikasi berbasis aturan tunggal. Dalam domain keamanan jaringan, *Random Tree* menunjukkan ketahanan terhadap data yang tidak seimbang dan menghasilkan hasil klasifikasi yang stabil [14].

Ketiga algoritma ini kemudian akan diterapkan pada dataset lalu lintas jaringan SDN untuk mengukur performa deteksi terhadap serangan DDoS berdasarkan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Penggunaan kombinasi algoritma yang berbeda ini diharapkan dapat memberikan gambaran menyeluruh mengenai efektivitas pendekatan klasifikasi dalam konteks deteksi dini serangan pada arsitektur jaringan yang bersifat terpusat seperti SDN. Inisialisasi model untuk masing-masing algoritma dilakukan secara terpisah sebagai berikut.

1) K-Nearest Neighbor (K-NN)

Untuk algoritma *K-Nearest Neighbor* (K-NN), parameter utama yang perlu diinisialisasi adalah nilai k yang menentukan jumlah tetangga terdekat yang akan dipertimbangkan dalam klasifikasi. Eksperimen awal dilakukan dengan berbagai nilai k (3, 5, 7, 9, 11) untuk menentukan nilai optimal melalui cross-validation pada data pelatihan. Metric distance yang digunakan adalah *Euclidean distance* yang sesuai untuk fitur numerik yang telah dinormalisasi.

2) Artificial Neural Network (A-NN)

Artificial Neural Network (A-NN) diinisialisasi dengan arsitektur *multilayer perceptron* yang terdiri dari input layer dengan 22 *neuron* (sesuai jumlah fitur setelah *preprocessing*), hidden layer dengan jumlah neuron yang akan dioptimasi, dan output layer dengan 1 neuron untuk klasifikasi biner. Fungsi aktivasi yang digunakan adalah ReLU untuk hidden layer dan sigmoid untuk output layer. Learning rate, batch size, dan jumlah epoch diinisialisasi dengan nilai standar yang kemudian akan disesuaikan melalui hyperparameter tuning.

3) Random Tree

Random Tree diinisialisasi dengan parameter default yang mencakup jumlah pohon (n_estimators), kedalaman maksimum pohon (max_depth), dan jumlah fitur yang dipertimbangkan untuk setiap split (max_features). Kriteria split menggunakan Gini impurity untuk mengukur kualitas pemisahan pada setiap node. Random state ditetapkan untuk memastikan reproducibility hasil, sementara parameter seperti min_samples_split dan min_samples_leaf diatur untuk mencegah overfitting.

Setiap model diinisialisasi dengan menggunakan library yang sesuai dan parameter yang telah dioptimasi. Proses inisialisasi juga mencakup pengaturan *environment* dan *dependency* yang diperlukan, serta validasi bahwa semua komponen model dapat berfungsi dengan baik menggunakan subset kecil dari data pelatihan. Logging dan monitoring sistem disiapkan untuk melacak performa dan progress pelatihan setiap model.

E. Pelatihan dan Pengujian Model

Pelatihan model dilakukan secara berurutan untuk ketiga algoritma menggunakan data pelatihan yang telah dipisahkan sebelumnya. Algoritma *K-Nearest Neighbor* tidak memerlukan proses pelatihan dalam arti konvensional karena merupakan *lazy learning algorithm*, namun proses fitting dilakukan untuk menyimpan data pelatihan dan melakukan optimasi parameter k melalui grid search dengan cross-validation. Proses ini menggunakan *k-fold cross-validation* dengan k=5 untuk mendapatkan estimasi performa yang robust dan menentukan nilai k optimal.

Pelatihan *Artificial Neural Network* dilakukan dengan menggunakan backpropagation algorithm dan optimizer Adam untuk mengoptimalkan bobot dan bias jaringan. Proses pelatihan dimonitor menggunakan training dan validation loss untuk mendeteksi overfitting. Early stopping mechanism diterapkan untuk menghentikan pelatihan jika validation loss tidak mengalami perbaikan dalam beberapa epoch berturut-turut. Batch training digunakan dengan ukuran batch yang dioptimalkan untuk keseimbangan antara kecepatan konvergensi dan stabilitas pelatihan.

Random Tree dilatih menggunakan *ensemble learning approach* dimana *multiple decision trees* dibangun secara paralel dengan bootstrap sampling dari data pelatihan. Setiap pohon dilatih dengan subset fitur yang dipilih secara acak untuk meningkatkan diversitas ensemble. Proses pelatihan menggunakan *out-of-bag* (OOB) scoring untuk estimasi performa internal tanpa memerlukan validation set terpisah. Feature importance dihitung selama proses pelatihan untuk memberikan insight tentang kontribusi setiap fitur dalam klasifikasi.

Pengujian model dilakukan setelah proses pelatihan selesai menggunakan data pengujian yang tidak pernah terlihat oleh model selama fase pelatihan. Setiap model melakukan prediksi pada seluruh data pengujian, dan hasil prediksi dibandingkan dengan label yang sebenarnya untuk menghitung berbagai matrik evaluasi. Proses pengujian juga mencakup analisis waktu inference untuk mengukur kecepatan prediksi setiap algoritma, yang penting untuk aplikasi real-time dalam deteksi serangan DDoS.

F. Evaluasi Model

Evaluasi model dilakukan menggunakan berbagai metrik klasifikasi yang sesuai untuk masalah deteksi anomali seperti serangan DDoS. Confusion matrix dibuat untuk setiap algoritma untuk memberikan gambaran detail tentang true positive, true negative, false positive, dan false negative. Dari confusion matrix ini, metrik dasar seperti accuracy, precision, recall, dan F1-score dihitung untuk memberikan evaluasi komprehensif tentang performa setiap model.

1) Akurasi (Accuracy)

Akurasi mengukur proporsi prediksi yang benar dari total seluruh prediksi [15]. Persamaan (1) adalah cara menghitung akurasi.

$$Accuracy = \frac{(TP+TN)}{TP+TN+FP+FN} \quad (1)$$

2) Presisi (Precision)

Presisi mengukur proporsi prediksi positif yang benar dari semua prediksi positif [16]. Persamaan (2) adalah cara menghitung presisi.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

3) Recall (Sensitivity / True Positive Rate)

Recall mengukur seberapa banyak kasus positif yang berhasil terdeteksi dari seluruh kasus positif yang sebenarnya [15]. Persamaan (3) adalah cara menghitung recall.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

4) F1-Score

F1-Score adalah rata-rata harmonik dari precision dan recall [17], Persamaan (4) adalah cara menghitung F1-Score.

$$F1\text{-}Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \quad (4)$$

5) AUC-ROC (Area Under Curve - Receiver Operating Characteristic)

AUC-ROC mengukur kemampuan model membedakan antara kelas Normal dan DDoS pada berbagai threshold [18]. Persamaan (5) dan (6) adalah cara menghitung TPR dan FPR.

$$TPR = \frac{TP}{TP+FN} \quad (5)$$

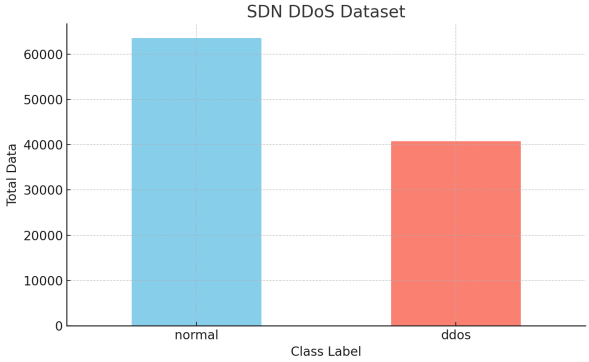
$$FPR = \frac{FP}{FP+TN} \quad (6)$$

6) Precision-Recall Curve

Precision-Recall Curve menunjukkan kurva hubungan antara precision dan recall di berbagai threshold. Metrik ini memberikan informasi lebih relevan dibanding ROC pada dataset dengan class imbalance [17].

3. HASIL DAN ANALISIS

Sebelum dilakukan pengujian dari masing-masing metode, penulis melakukan visualisasi terhadap data dari SDN DDoS Dataset yang ditunjukkan pada Gambar 2. Distribusi data antara normal dan DDoS masih terdapat ketidakseimbangan kelas dengan rasio normal dengan DDoS sekitar 3:2. Namun, data ini cukup representatif dengan jumlah DDoS lebih besar dari 100 ribu data sama dengan pada penelitian sebelumnya [8].



Gambar 2. Perbandingan dataset antara kelas normal dan DDoS

Setelah mengetahui kecukupan data SDN DDoS, penulis melakukan pemodelan untuk membandingkan ketiga jenis metode yaitu KNN, ANN, dan *Random Tree*. Perbandingan ini merujuk pada evaluasi dengan menggunakan empat metrik utama: akurasi, presisi, recall, dan F1-Score. Secara keseluruhan, metode ANN menghasilkan nilai yang terbaik yaitu akurasi 96.85%, presisi 94.35%, recall 97.79%, dan F1-Score 96.04% yang ditunjukkan pada Tabel 2. Sedangkan nilai yang terendah didapatkan oleh *Random Tree* akurasi 86.49%, presisi 89.42%, recall 74.22%, dan F1-Score 81.12%.

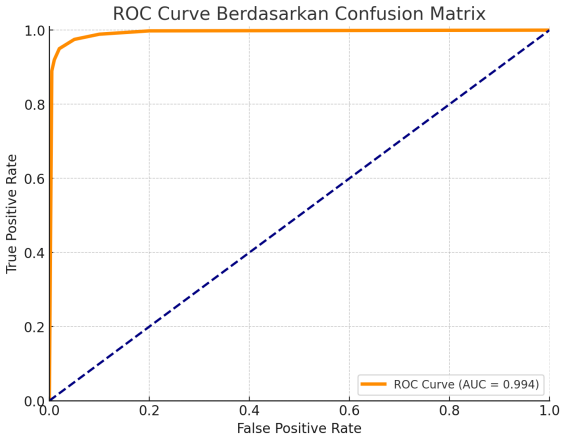
Tabel 2. Komparasi Performa tiap model

Matriks	***Random Tree* (Gini Index)**	**K-NN**	**A-NN**
Akurasi	86.49%	88.89%	96.85%
Presisi	89.42%	88.24%	94.35%
Recall	74.22%	82.57%	97.79%
F1-Score	81.12%	85.31%	96.04%
Confusion Matriks	**17994,3154, 1074,9081**	**17722,2132, 1346,10103**	**18352,270, 716,11965**

Berdasarkan data pada tabel 2 di atas model *Random Tree* menghasilkan akurasi 86,49%, presisi 89,42%, recall 74,22%, dan F1-score 81,12%. *Confusion matrix* menunjukkan bahwa model ini benar mengklasifikasikan 17.994 data normal dan 1.074 serangan, namun salah mengklasifikasikan 3.154 data normal dan 9.081 data serangan. *Random Tree* merupakan algoritma berbasis decision tree yang menggunakan Gini Index sebagai kriteria pemilihan split. *Gini Index* memprioritaskan pemisahan data berdasarkan ketidakmurnian (impurity) antar kelas, namun algoritma ini rentan terhadap *overfitting* pada data pelatihan, terutama jika pohon dibangun terlalu dalam. Selain itu, Gini Index juga tidak seefektif entropy dalam menghadapi ketidakseimbangan kelas, yang bisa menjelaskan rendahnya nilai recall pada model ini. Kelemahan dalam mengidentifikasi serangan (*true positive*) mengindikasikan bahwa model ini belum mampu menangkap seluruh variasi pola serangan DDoS.

Algoritma K-NN memperoleh nilai akurasi sebesar 88,89%, presisi 88,24%, recall 82,57%, dan F1-score 85,31%. Dari *confusion matrix*, diketahui bahwa model ini berhasil mengklasifikasikan 17.722 data normal dengan benar (*True Negative*), tetapi juga mengklasifikasikan secara salah 2.132 data normal sebagai serangan (*False Positive*). Model ini mampu mengenali 1.346 serangan dengan benar (*True Positive*), tetapi gagal mendeteksi 10.103 serangan (*False Negative*). Secara teoritis, algoritma K-NN merupakan model non-parametrik berbasis instance yang melakukan klasifikasi berdasarkan kemiripan fitur dengan tetangga terdekat dalam ruang vektor. Dalam konteks dataset besar dan kompleks seperti data serangan DDoS, K-NN dapat mengalami kelemahan dalam menghadapi distribusi kelas yang tidak seimbang, serta kesulitan dalam menggeneralisasi data yang memiliki noise atau fitur yang saling tumpang tindih. Hal ini dapat menjelaskan tingginya jumlah *false negative* pada model ini.

Sedangkan Model A-NN menunjukkan performa terbaik, dengan akurasi mencapai 96,85%, presisi 94,35%, recall 97,79%, dan F1-score 96,04%. Berdasarkan confusion matrix, model ini memiliki tingkat kesalahan yang sangat rendah, hanya 270 false positive dan berhasil mengklasifikasikan 716 serangan (True Positive) serta 18.352 data normal (*True Negative*). Walaupun angka *False Negative* relatif besar (11.965), A-NN tetap menunjukkan keseimbangan yang sangat baik antara presisi dan recall. Kinerja superior dari A-NN dapat dijelaskan melalui sifat arsitekturnya yang mampu mempelajari representasi non-linier kompleks dari data melalui lapisan tersembunyi. *Neural network* memiliki kemampuan untuk menangkap pola yang sulit ditangkap oleh model konvensional seperti K-NN atau *decision tree*. A-NN juga memiliki keunggulan dalam melakukan feature extraction otomatis dan mampu beradaptasi terhadap kompleksitas hubungan antar fitur dalam dataset, termasuk pola lalu lintas jaringan pada serangan DDoS yang tidak selalu linear. Hal ini menunjukkan metode A-NN dapat melakukan pemodelan pola yang lebih kompleks dan non-linier dalam *traffic* DDoS. Dari 22 fitur yang digunakan metode ANN menunjukkan data yang heterogen sehingga ANN mampu untuk mengekstraksi fitur-fitur yang dengan kemampuan pembelajaran yang lebih mendalam. Data-data yang outliers juga akan lebih diatasi oleh ANN dibandingkan KNN maupun *Random Tree*. Ketika adanya peningkatan suatu fitur maka KNN akan terjadi curse of dimensionality yaitu penurunan performa ketika dimensi fitur besar. Sedangkan pada *Random Tree* belum dapat menangkap kompleksitas model dimana fitur tidak hanya berupa kategori diskrit melainkan *continous* yang akan membuat keputusan non-linier yang sulit dibandingkan A-NN.



Gambar 3 ROC Curve (model terbaik A-NN)

Berdasarkan Grafik *Receiver Operating Characteristic* (ROC) yang ditampilkan pada gambar 3 diatas yakni merupakan hasil evaluasi performa model klasifikasi berbasis *Artificial Neural Network* (A-NN) dalam membedakan antara kelas positif (serangan DDoS) dan kelas negatif (normal). Kurva ROC (berwarna merah) menunjukkan hubungan antara *True Positive Rate* (TPR) atau sensitivitas terhadap *False Positive Rate* (FPR) pada berbagai nilai ambang klasifikasi (*threshold*). Dari grafik tersebut, A-NN menghasilkan nilai AUC (*Area Under Curve*) sebesar 0.994, yang menunjukkan bahwa model ini memiliki performa yang sangat tinggi dan hampir sempurna dalam melakukan klasifikasi. Secara teoritis, nilai AUC yang mendekati 1 menandakan bahwa model memiliki kemampuan sangat baik dalam membedakan antara dua kelas, dan dalam konteks ini, ANN terbukti mampu mengidentifikasi serangan DDoS secara sangat akurat.

Karakteristik kurva yang tampak melengkung tajam ke arah kiri atas grafik menunjukkan bahwa model A-NN dapat mendeteksi sebagian besar serangan DDoS (*true positive*) dengan tingkat kesalahan klasifikasi yang sangat rendah (*false positive*). Ini mencerminkan bahwa A-NN tidak hanya mampu mengenali pola serangan secara efektif, tetapi juga menjaga ketelitian dalam menghindari kesalahan mendeteksi trafik normal sebagai serangan. Karakteristik ini sangat penting dalam sistem keamanan jaringan, di mana kecepatan dan ketepatan dalam mendeteksi ancaman menjadi prioritas utama. Selain itu, grafik juga memperlihatkan kurva threshold (berwarna biru) yang menggambarkan bagaimana perubahan nilai ambang klasifikasi mempengaruhi performa model. Ketika threshold diturunkan, model menjadi lebih sensitif dalam mengenali kelas positif, namun berpotensi meningkatkan tingkat false positive. Oleh karena itu, dalam implementasi praktis, pemilihan nilai threshold perlu disesuaikan dengan konteks operasional sistem. Dalam kasus deteksi serangan DDoS, trade-off ini harus dikelola dengan baik karena mendeteksi serangan lebih penting daripada menghindari false alarm, terutama dalam sistem real-time yang memprioritaskan mitigasi ancaman.

Secara keseluruhan, nilai AUC sebesar 0.994 yang dihasilkan oleh ANN menunjukkan bahwa pendekatan ini sangat efektif dalam mengklasifikasikan trafik jaringan dan mendeteksi serangan DDoS. Hasil ini sejalan dengan keunggulan ANN dalam mengenali pola non-linear dan kompleksitas data yang tinggi, sehingga ANN sangat cocok digunakan pada domain keamanan jaringan, khususnya untuk sistem deteksi intrusi berbasis pembelajaran mesin (*Machine Learning*).

4. KESIMPULAN

Penelitian ini berhasil membandingkan tiga metode pembelajaran mesin untuk deteksi serangan DDoS menggunakan SDN DDoS Dataset dan menunjukkan bahwa *Artificial Neural Network* (ANN) secara signifikan mengungguli metode lainnya dengan mencapai akurasi 96,85%, presisi 94,35%, recall 97,79%, F1-Score 96,04%, dan AUC 0,994, dibandingkan dengan *K-Nearest Neighbor* (K-NN) yang mencapai akurasi 88,89% dan *Random Tree* dengan akurasi terendah 86,49%. Performa superior A-NN disebabkan oleh kemampuannya menangkap pola non-linear kompleks melalui arsitektur jaringan syaraf berlapis, melakukan feature extraction otomatis dari 22 fitur heterogen, serta beradaptasi lebih baik terhadap kompleksitas hubungan antar fitur dan data outliers dibandingkan metode konvensional. Hasil ROC curve dengan AUC mendekati sempurna menunjukkan bahwa A-NN mampu mendeteksi sebagian besar serangan DDoS dengan tingkat kesalahan sangat rendah, menjadikannya pilihan ideal untuk implementasi sistem deteksi intrusi real-time dalam lingkungan *Software Defined Network*.

REFERENSI

[1] A. P. Segara, R. M. Ijtihadie, and T. Ahmad, "IMPLEMENTASI ALGORITMA SHORTEST PATH JOHNSON UNTUK MEKANISME ROUTE DISCOVERY PADA SOFTWARE DEFINED NETWORK," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 16, no. 1, p. 10, Jan. 2021, doi: https://doi.org/10.12962/j24068535.v19i1.a1011.

[2] A. Zaman, S. A. Khan, N. Mohammad, A. A. Ateya, S. Ahmad, and M. A. ElAffendi, "Distributed Denial of Service Attack Detection in Software-Defined Networks Using Decision Tree Algorithms," *Future Internet*, vol. 17, no. 4, Apr. 2025, doi: 10.3390/fi17040136.

[3] M. E. Haque, A. Hossain, M. S. Alam, A. H. Siam, S. M. F. Rabbi, and M. M. Rahman, "Optimizing DDoS Detection in SDNs Through Machine Learning Models," in *Proceedings - 2024 IEEE 16th International Conference on Communication Systems and Network Technologies, CICN 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 426–431. doi: 10.1109/CICN63059.2024.10847458.

[4] A. Hamarshe, H. I. Ashqar, and M. Hamarsheh, "Detection of DDoS Attacks in Software Defined Networking Using Machine Learning Models," May 2023. doi: https://doi.org/10.1007/978-3-031-33743-7_51.

[5] Z. Ma and B. Li, "A DDoS attack detection method based on SVM and K-nearest neighbour in SDN environment 'A DDoS attack detection method based on SVM and K-nearest neighbour in SDN environment," Oct. 2020. doi: https://doi.org/10.1504/IJCSE.2020.111431.

[6] B. Nuralamsyah, S. R. Anggraeni, L. Awwabi, N. A. Ranggianto, H. Studiawan, and A. M. Shiddiqi, "Performance Analysis Between EOTI-K-Means++, EOTI, and KNN for Brute Force Detection System," in *2022 10th International Conference on Information and Communication Technology (ICoICT)*, IEEE, Aug. 2022, pp. 53–58. doi: 10.1109/ICoICT55009.2022.9914878.

[7] W.-W. Tay, S.-C. Chong, and L.-Y. Chong, "DDoS Attack Detection with Machine Learning," *Journal of Informatics and Web Engineering*, vol. 3, no. 3, pp. 190–207, Oct. 2024, doi: 10.33093/jiwe.2024.3.3.12.

[8] F. Ferdiansyah, D. Antoni, M. Valdo, M. Mikko, C. Mukmin, and U. Ependi, "Machine Learning Models for DDoS Detection in Software-Defined Networking: A Comparative Analysis," *Journal of Information Systems and Informatics*, vol. 6, no. 3, pp. 1790–1803, Sep. 2024, doi: 10.51519/journalisi.v6i3.864.

[9] K. M. Ko, J. M. Baek, B. S. Seo, and W. B. Lee, "Comparative Study of AI-Enabled DDoS Detection Technologies in SDN," *Applied Sciences (Switzerland)*, vol. 13, no. 17, Sep. 2023, doi: 10.3390/app13179488.

[10] A. Prasad, S. Prasad, K. Arockiasamy, and X. Yuan, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Detection of DDoS Attack in Software-Defined Networking Environment and Its Protocol-wise Analysis using Machine Learning," Sep. 2022. [Online]. Available: www.ijisae.org

[11] G. Gnana Priya, S. H. Shriram, S. Jeeva, G. Sakthi Priya, and K. Balasubadra, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Detection of Distributed Denial of Service (DDOS) Attack Using Logistic Regression and K Nearest Neighbor Algorithms," Jan. 2024. [Online]. Available: http://www.mathtype.com

[12] S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNN with the Degree of DDoS Attack in Software-Defined Networks," *IEEE Access*, vol. 8, pp. 5039–5048, 2020, doi: 10.1109/ACCESS.2019.2963077.

[13] N. J. Schaub and N. Hotaling, "Assessing Efficiency in Artificial Neural Networks," *Applied Sciences (Switzerland)*, vol. 13, no. 18, Sep. 2023, doi: 10.3390/app131810286.

[14] A. Ekawijana, A. Bakhrun, and M. T. Kurniawan, "Deteksi Serangan DDOS Pada Jaringan SDN dengan Metode Random Forest," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 1, p. 685, Jan. 2024, doi: 10.30865/mib.v8i1.6928.

[15] J. Han, M. Kamber, and J. Pei, "Data Mining. Concepts and Techniques, 3rd Edition (The Morgan Kaufmann Series in Data Management Systems)," 2011.

[16] C. C. Aggarwal, *Data Mining*. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-14142-8.

[17] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS One*, vol. 10, no. 3, Mar. 2015, doi: 10.1371/journal.pone.0118432.

[18] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit Lett*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.